

Московский государственный технический университет им. Н.Э. Баумана  
Факультет «Информатика и системы управления»  
Кафедра «Системы обработки информации и управления»



**Отчет**  
**РК № 2**  
**По курсу «РИП»**

**Вариант 10**

**ИСПОЛНИТЕЛЬ:**

Группа ИУ5-53  
Дудник М.В.

**ПРЕПОДАВАТЕЛЬ:**

Гапанюк Ю.Е.

---

Москва 2021

## Задание

Рубежный контроль представляет собой разработку веб-приложения с использованием фреймворка Django. Веб-приложение должно выполнять следующие функции:

1. Создайте проект Python Django с использованием стандартных средств Django.
2. Создайте модель Django ORM, содержащую две сущности, связанные отношением один-ко-многим в соответствии с Вашим вариантом из условий рубежного контроля №1.
3. С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
4. Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

## Код программы

models.py

```
from django.db import models

class Browsers(models.Model):
    id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=20)

    class Meta:
        db_table = 'browsers'

class PC(models.Model):
    id = models.IntegerField(primary_key=True)
    memory = models.IntegerField()
    brand = models.CharField(max_length=20)
    browser_id = models.ForeignKey(Browsers, on_delete=models.PROTECT)

    class Meta:
        db_table = 'pc'
```

## serializers.py

```
1 from rest_framework import serializers
2
3 from browsersPc.models import Browsers, PC
4
5 class PCSerializer(serializers.ModelSerializer):
6     class Meta:
7         model = PC
8         fields = "__all__"
9
10 class BrowsersSerializer(serializers.ModelSerializer):
11     class Meta:
12         model = Browsers
13         fields = "__all__"
```

## views.py

```
1 from django.shortcuts import render
2 from rest_framework import viewsets
3
4 from browsersPc.models import PC, Browsers
5 from browsersPc.serializers import PCSerializer, BrowsersSerializer
6
7 class PCViewSet(viewsets.ModelViewSet):
8     queryset = PC.objects.all()
9     serializer_class = PCSerializer
10
11 class BrowsersViewSet(viewsets.ModelViewSet):
12     queryset = Browsers.objects.all()
13     serializer_class = BrowsersSerializer
14
15 def report(request):
16     print(PC.objects.select_related('browser_id')[0].memory)
17     return render(request, 'report.html', {'data': {
18         'pc': PC.objects.select_related('browser_id')
19     }})
```

## urls.py

```
from rest_framework import routers
from django.urls import path, include
from django.contrib import admin
from django.urls import path
from browsersPc import views as views

router = routers.DefaultRouter()
router.register('pc', views.PCViewSet)
router.register('browsers', views.BrowsersViewSet)

urlpatterns = [

    path('', include(router.urls)),
    path('report/', views.report),
    path('api-auth/', include('rest_framework.urls', namespace='rest_framework')),
    path('admin/', admin.site.urls),
```

## report.html

```
{% extends 'base.html' %}

{% block title %}Компьютеры-браузеры{% endblock %}

{% block content %}
    <div>
        <h1>Список компьютеров:</h1>
    </div>
    <ul>
        {% for pc in data.pc %}
            <li>Размер памяти: {{ pc.memory }}</li>
            <li>Бренд: <i>{{ pc.brand }}</i></li>
            <li>Браузер: <i>{{ pc.browser_id.name }}</i></li>
            <br>
            {% empty %}
                <li>Список пуст</li>
            {% endfor %}
        </ul>
    </ul>
{% endblock %}
```

## Запросы к бд

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```

```
[
  {
    "id": 1,
    "memory": 16,
    "brand": "Dell",
    "browser_id": 1
  },
  {
    "id": 2,
    "memory": 8,
    "brand": "Asus",
    "browser_id": 2
  },
  {
    "id": 3,
    "memory": 4,
    "brand": "Honor",
    "browser_id": 2
  },
  {
    "id": 4,
    "memory": 16,
    "brand": "Apple",
    "browser_id": 1
  },
  {
    "id": 5,
    "memory": 8,
    "brand": "HP",
    "browser_id": 4
  }
]
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
```

```
[
  {
    "id": 1,
    "name": "safari"
  },
  {
    "id": 2,
    "name": "chrome"
  },
  {
    "id": 3,
    "name": "mozilla"
  },
  {
    "id": 4,
    "name": "yandex"
  }
]
```

### Результат работы

← → ↻ ⓘ 127.0.0.1:8000/report/

## Список компьютеров:

- Размер памяти: 16
- Бренд: *Dell*
- Браузер: *safari*
- Размер памяти: 8
- Бренд: *Asus*
- Браузер: *chrome*
- Размер памяти: 4
- Бренд: *Honor*
- Браузер: *chrome*
- Размер памяти: 16
- Бренд: *Apple*
- Браузер: *safari*
- Размер памяти: 8
- Бренд: *HP*
- Браузер: *yandex*